THIRD PERSON TRACKING CAMERA

version 1.0

nyangireworks@gmail.com

Table of Contents

1. Introduction

Thank you for purchasing the *Third Person Tracking Camera*!

If you have any issues, questions or bugs to report, suggestions, feature requests don't hesitate to contact me via email:

nyangireworks@gmail.com

This document will walk you through setting up the camera with Unity3D, along with explaining its features and properties.

The camera can be setup without any C# knowledge.

This document assumes a basic knowledge of Unity3D.

2. Brief overview

The ***Third Person Tracking Camera*** is a collection of C# scripts and prefabs to be used to set up a camera that automatically tracks a subject (gameobject) over a predetermined path during gameplay in realtime.

The tracking camera uses a user defined curve path (spline; bezier curve) or point and automatically follows a subject as they move through the scene keeping them in the center or within margins of the screen.

It can be used to automatically control the cameras flow, framing, position and FOV (field of view) in realtime during gameplay without any player control to set up predetermined paths for the camera throughout gameplay.

The camera can be used in conjunction with other camera scripts for specific areas or it can be used solo throughout the game.

The asset consists of prefabs meant to be used to set up the camera and its behavior. The scripts that control the camera, prefabs and editor. A demo scene showing various uses of the camera and examples. And other assets needed for the Demo.

The *ThirdPersonTrackingCamera* prefab (or script) is necessary to be on a camera in order for the functionality. *CameraSpline* and *CameraFixedLocation* prefabs serve to set up the paths and points that acts as constraints for the camera and the *CameraSwitchTrigger* prefab is a trigger volume used to switch between different paths and points.

There are two main ways the camera can be set up in: *Fixed* mode and *Bezier Follow* Mode

The ***Fixed*** mode is the simplest mode that uses a fixed point in the world and a predetermined angle.

In the Fixed mode the Camera can be completely fixed to the position and orientation of the point, and can be set to rotate towards the subject and have a custom FOV value.

The ***Bezier Follow*** mode follows the player/object on a bezier curve in realtime by determining the best location on the curve to keep the player centered. It can follow the orientation of its points exactly or freely rotate the camera on any point on the curve and it can animate the cameras FOV as it moves over the curve.

Additionally instead of tracking a subject it can track another linear path allowing for more control in certain scenarios.

3. Quick Setup

This part will walk you through the minimal steps necessary to have a functioning camera in your scene.

Refer to next chapters for a more in depth explanation and setup of advanced features.

You can use the demo scene provided to see the example setups.

3.1. Setting Up the Camera Script

Delete the Main Camera and drag the *ThirdPersonTrackingCamera* prefab from the prefabs folder into your scene. OR If you have a Camera and wish to add the Third Person Tracking functionality drag the *ThirdPersonTrackingCamera* script from the Scripts folder onto your existing camera.

Afterwards make sure to assign the *gameobject* the camera will track into the Follow field.

The camera itself wont do anything until a curve or point is set up and the camera is assigned to it using a trigger or script.

3.2. Setting up the Camera Bezier Curves

Drag the *CameraSpline* prefab onto the scene. The spline is a visualization of the path the camera is going to be constrained to. Selecting a point allows manipulation of the point. To visualize the camera at certain points press the **Align main Camera to selected** and look at the game view. To place the point where the editor/scene camera press **Align Selected To Scene Camera**.

For full instructions on how to manipulate the bezier curve editor refer to section 4. Bezier Curve Editor Controls.

In order for the camera to start on the bezier curve on pressing play drag the *CameraStart* script onto a *gameobject* that’s present in the scene and drag the *CameraSpline* onto the *StartBezier* property.

3.3. [Optional] Setting up the Camera Fixed point

Drag The *CameraFixedLocation* prefab into the scene, it is marked with a diamond.

Set your editor/scene camera to a desired location and select the *CameraFixedLocation* in your scene. Go to the *GameObject* dropdown and select Align With View.

In order to preview the orientation of the Fixed Point go to the *GameObject* dropdown and select Align View To Selected.

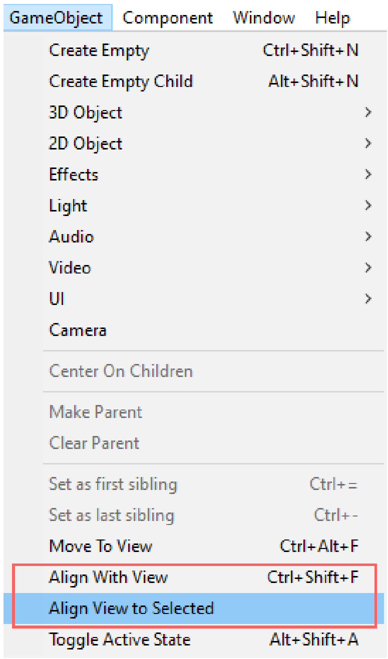

GameObject
Component
Window
Help
Create Empty Ctrl+Shift+N
Create Empty Child Alt+Shift+N
3D Object
2D Object
Effects
Light
Audio
Video
UI
Camera
Center On Children
Make Parent
Clear Parent
Set as first sibling Ctrl+=
Set as last sibling Ctrl+-
Move To View Ctrl+Alt+F
Align With View Ctrl+Shift+F
Align View to Selected
Toggle Active State Alt+Shift+A

Figure 1: Unity Gameobject dropdown

In order for the camera to start at that location when pressing play drag the *CameraStart* Script onto a gameobject thats present in the scene and drag the *CameraFixedLocation* from the scene onto the Start Bezier property (note: both *CameraFixedLocation* and *CameraSpline* are compatible with the Start Bezier field).

3.4. Setting Up The Camera Switch Volumes

Drag the CameraSwitchTrigger prefab onto the scene. The CameraSwitchTrigger is a volume that switches the camera to a different Spline or Fixed Point upon detecting that a game object tagged “Player” entered the volume.

Drag the CameraSpline or CameraFixedLocation gameobject into the Spline property, thats the spline the trigger will switch to.

4. Bezier & Linear Path Editor

This asset contains its own bezier and linear paths used to control the camera.

The controls for the Linear path found in the LinearPath prefab are equivalent to the Bezier Curve controls with the exclusion of Bezier only features, so this section will only cover the Bezier curve editor as the Linear Path editor is equivalent.

4.1. Setup

The *CameraSpline* prefab contains the Bezier Curve scripts that create and display an editable bezier curve. Similar controls are used for the LinearPath prefab.

To edit the bezier curve the *CameraSpline* gameobject must be selected and **ONLY** the CameraSpline gameobject.

To select the game object simply click on the CameraSpline icon in the Editor or select the CameraSpline gameobject in the Hierarchy.

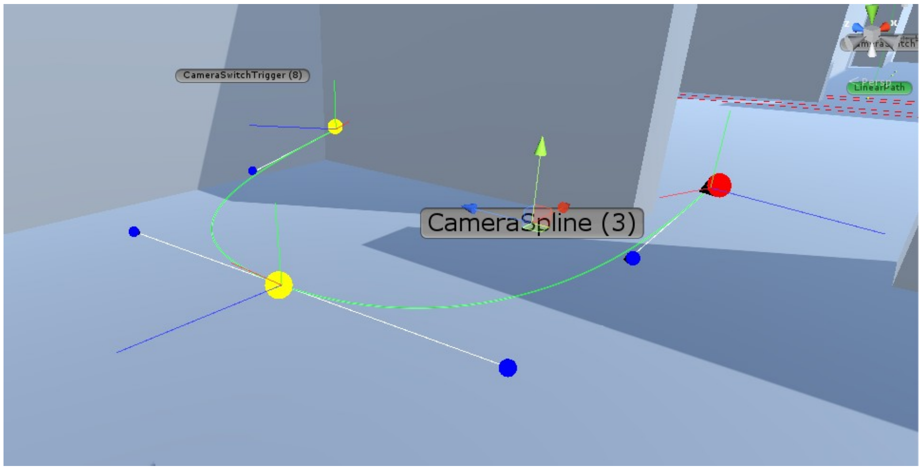


Figure 2: Selected CameraSpline

Once selected the gizmo shows up in the Editor Window and properties in the Inspector.

For more information on what each property does jump to section 6.1 Bezier Editor.

4.2. Controls:

Left Click – select (select Anchor and control points)

Left Click + Drag – move point (the position gizmo can also be used instead)

Shift+ Left Click – Add Anchor Point (Adds a new anchor point after the last anchor point)

Ctrl + Left Click – Insert Anchor Point (Split a segment by adding an anchor in the middle

Ctrl + Shift + Left Click – Delete Anchor Point

Ctrl+Z – Undo (Can Undo above changes along with point manipulation, create new button and other inspector changes that affect the bezier curve in the Editor)

The main control for the spline can move the entire spline.

To edit the points left click on a point while the CameraSpline is selected, a selected point show a bright blue color and additional inspector properties show up that control the values for that point.

There are two types of points: **Anchor points** are points through which the curve (yellow) and **control points** that control the flow of the curve (dark blue). Selecting the anchor points opens up their properties in the inspector (and highlights them in cyan), these properties belong to the individual anchor point. The red point is the starting point adding another point always connects the new point to the last free end point.

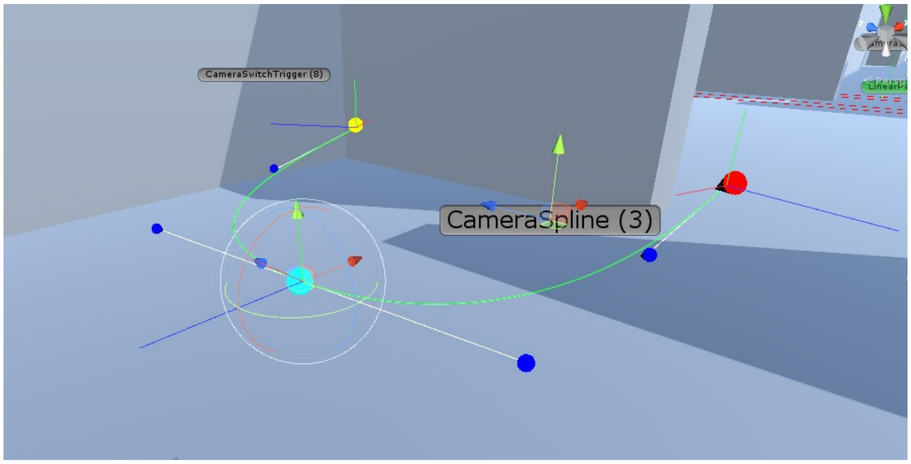
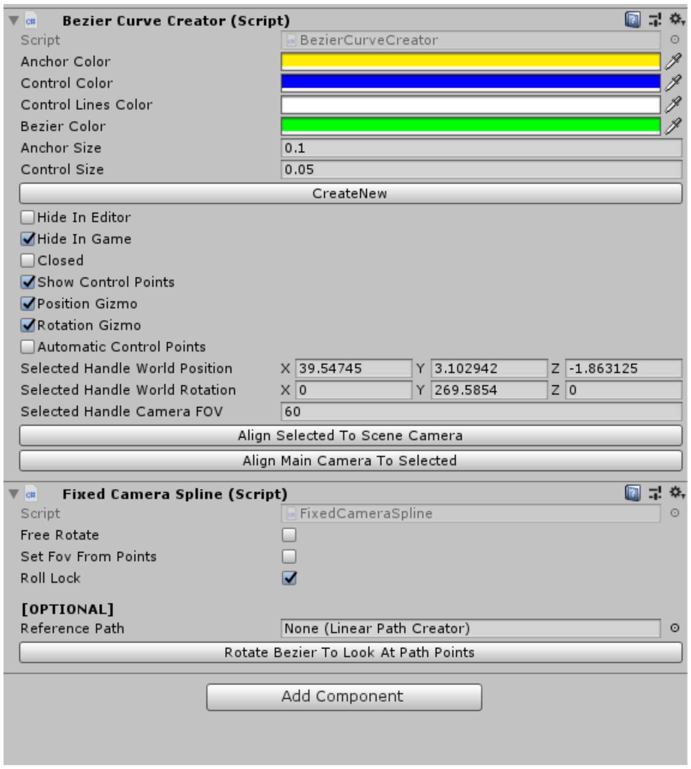


Figure 3: The selected anchor point turns cyan and shows the gizmo



Bezier Curve Creator (Script)

Script: BezierCurveCreator

Anchor Color: 🟨

Control Color: 🟦

Control Lines Color: ⬜

Bezier Color: 🟩

Anchor Size: 0.1

Control Size: 0.05

☐ Hide In Editor

☑ Hide In Game

☐ Closed

☑ Show Control Points

☑ Position Gizmo

☑ Rotation Gizmo

☐ Automatic Control Points

Selected Handle World Position: X 39.54745 Y 3.102942 Z -1.863125

Selected Handle World Rotation: X 0 Y 269.5854 Z 0

Selected Handle Camera FOV: 60

Fixed Camera Spline (Script)

Script: FixedCameraSpline

Free Rotate: ☐

Set Fov From Points: ☐

Roll Lock: ☑

[OPTIONAL]

Reference Path: None (Linear Path Creator)

CreateNew
Align Selected To Scene Camera
Align Main Camera To Selected
Rotate Bezier To Look At Path Points
Add Component

Figure 4: Inspector properties for the Spline and the selected anchor point

Rotating the anchor points rotates the position at which the camera will be positioned in that anchor point (unless free rotate is ticked, but the positioning algorithm still uses these directions to inform the cameras position on the curve so its advisable to edit them to point in the general direction where the camera should view).

The blue line is the direction the camera points in (local Z), the red line is the right side of the camera (local X) and the green line the up vector of the camera (local Y), the colors match the unity gizmo.

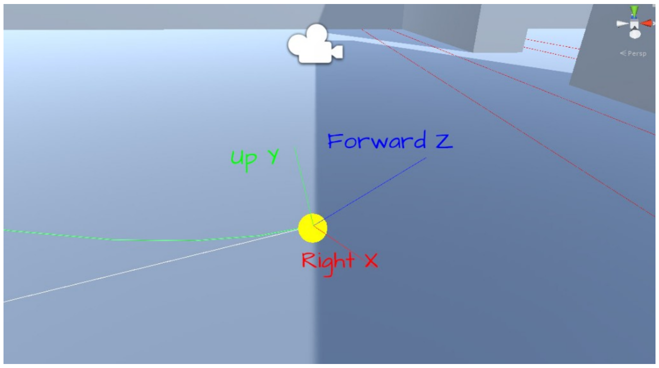


Figure 5: Anchor points have orientation that the camera uses

5. Features

There are two main modes the camera can be set up in: *Fixed* mode and *Bezier Follow* Mode.

Each of these modes have a variety of features to further setup the cameras behaviour.

The fixed mode behavior is tied to the *CameraFixedLocation* prefab, and the Bezier follow mode is tied to the *CameraSpline* prefab.

5.1. Fixed Mode & Automatic Rotate Mode

The *CameraFixedLocation* prefab is used for the Fixed and Automatic rotate modes.

The **Fixed** mode is the simplest mode that uses a fixed location and rotation for the camera, allowing for fixed angle shots.

The **Automatic rotate** mode can be enabled on the *CameraFixedLocation* by checking *Allow Rotation*. In this mode the camera remains at the fixed location but freely rotates to keep the subject within specified margins on the view of the camera.

5.2. Automatic Bezier Follow Mode

The main feature of this asset is the ability for the camera to smoothly position itself on a bezier curve path in realtime during gameplay to track a player controlled subject as they navigate the scene, aka the **Bezier Follow** mode.

This allows for cinematic non-player operated camera angles that smoothly follow the player throughout the scene.

The *CameraSpline* prefab is used for modes that follow a bezier curve. The prefab contains an editable bezier curve which the camera will be constrained to.

The tracking algorithm will automatically determine the best position on the curve in order to have the subject in center view and smoothly position there.

The tracking algorithm uses the values set on each anchor point – position, rotation (and FOV if enabled) and linearly interpolates these values along the curve up to the next anchor point. This means that as the camera moves along the path it smooths from one anchor points view to the next.

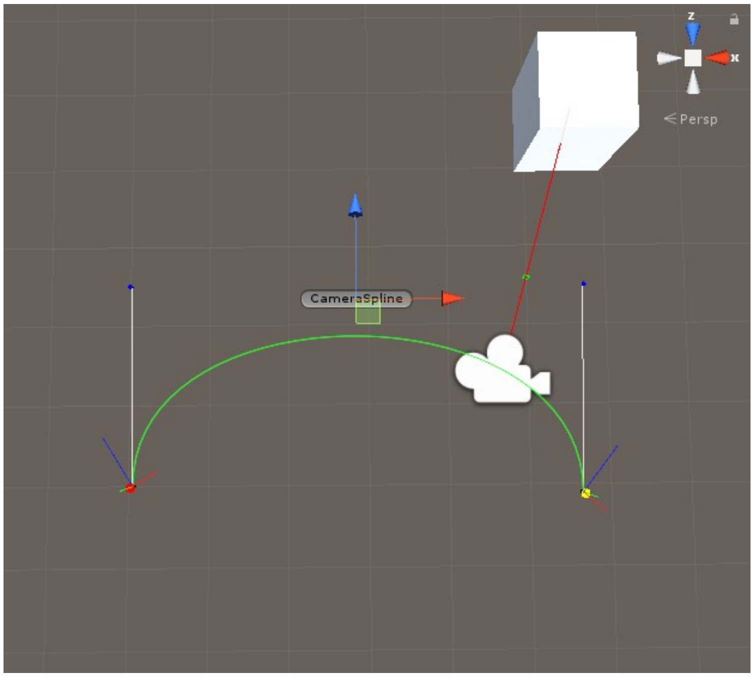


Figure 6: The red line is the cameras forward, a interpolated value from the left and right anchor point

The Tracking Algorithm determines the best position on the curve that keeps the subject in the frame, with this in mid its advisable in order to get the best behavior of the camera to tweak anchor rotation and position in a way that each anchor has a unique view as two points on the curve with similar viewpoints could lead to ambiguity.

Certain complex movements that do not correspond to the players positions may need the use of a **Linear Path** (section 5.6.).

Furthermore the *AllowRotation* option has no bearing on the placement of the camera on the curve, the placement is calculated from the orientation of the anchor points therefore they need to be properly set up even when using *AllowRotation*.

If the curve is more complex the Reference Path mode (section 5.6.) can be used to more finely control the positioning of the camera based on subject world position.

5.3. Free Rotate in Bezier follow

The Free rotate is an extra option on the *CameraSpline* prefab that allows the camera to freely rotate at any point on the curve to center the subject in frame.

Without the free rotate option the camera only uses angles set by the anchor points on the bezier curve, while this allows for precise control of angles during gameplay the free rotate mode follows the subject precisely allowing for a wider range of view.

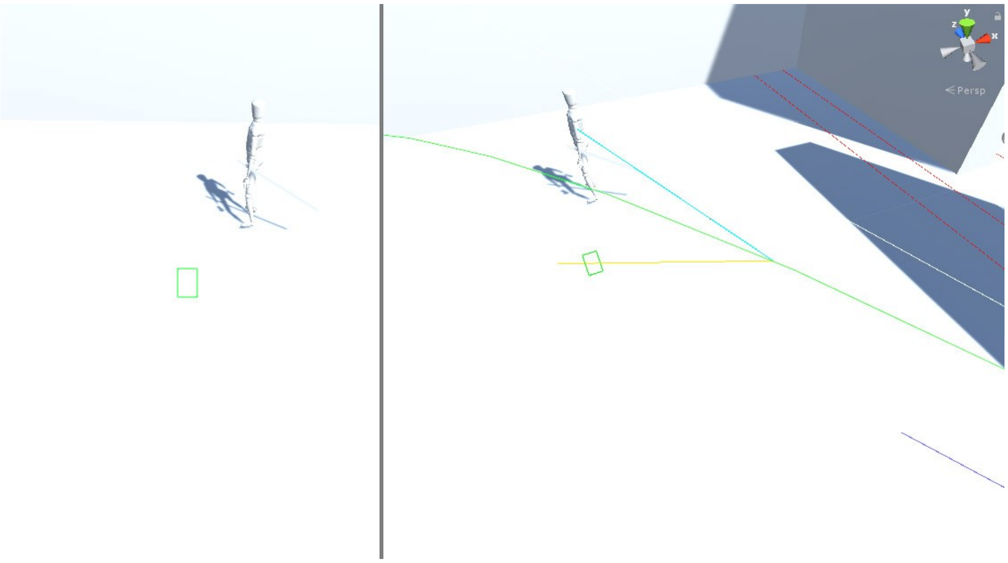


Figure 7: Without free rotate the camera matches the user defined angles on the curve while finding the optimal placement for the subject to be in frame

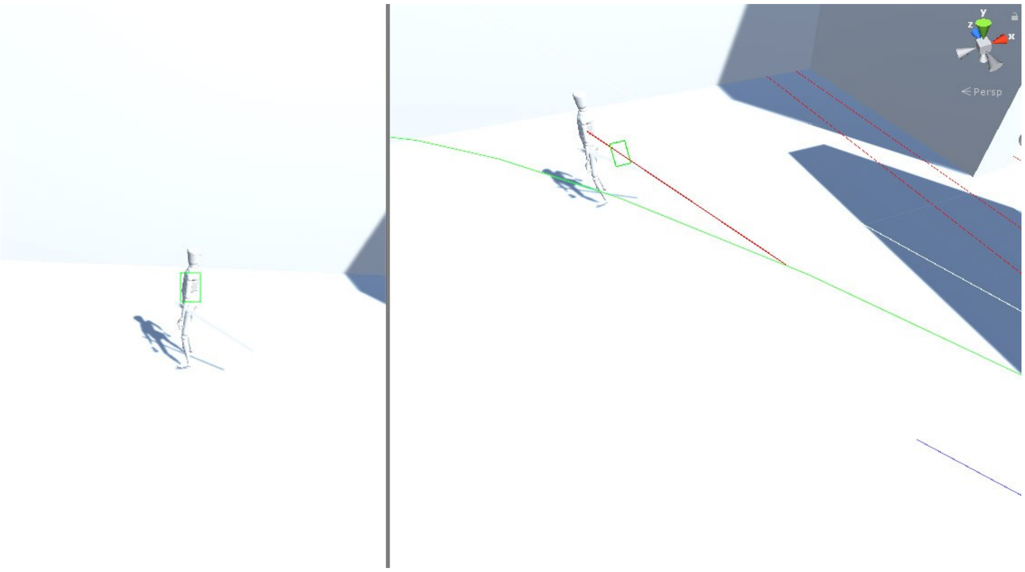


Figure 8: With free rotate once the camera also rotates towards the subject after placement

5.4. Set Fov From Points

Set FOV from points allows for overriding the cameras FOV and using anchor points to set a custom FOV value at any point in the curve. The FOV values will linearly interpolate across the curve between points.

5.5. Roll Lock

The Roll Lock locks the Z rotation of the camera (aka its viewing angle). This overrides roll set on Anchor points to ensure the camera is always horizontal to the ground and not rotated.

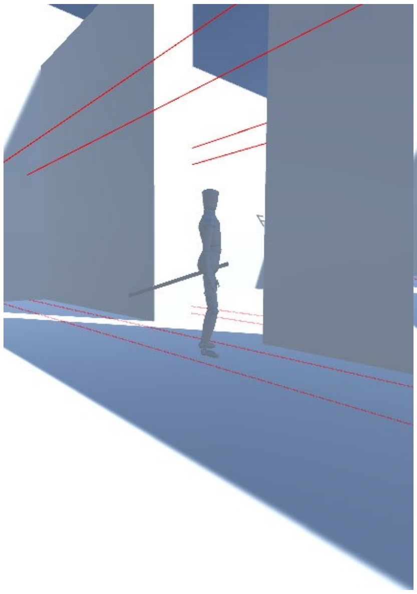


Figure 9: With roll lock the camera wil always stay level

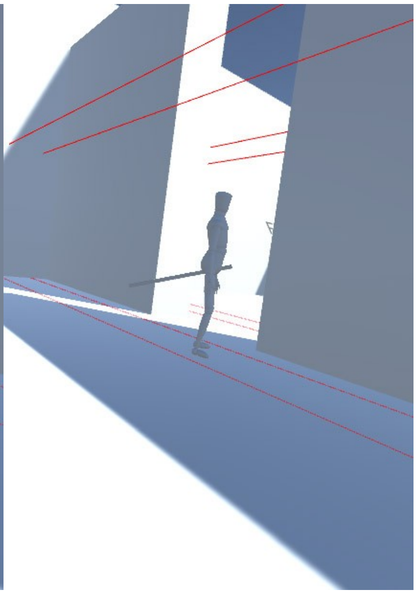


Figure 10: Without roll lock its possible to control the cameras angle using the anchor points

5.6. Bezier Follow With Reference Path Mode

The Reference path is an optional reference to the *LinearPath* prefab in the *CameraSpline* inspector.

The *LinearPath* prefab contains an editable linear path that can be setup in the scene.

If a reference is provided the camera will not use the default automatic tracking algorithm but will instead follow points on the linear path according to the closest point to the subject.

This is an additional functionality that allows for more complex behavior of the curve when the default tracking algorithm is not suited.

The linear paths points correspond to the points on the bezier curve, for a linear path to fully map to the bezier curve it should have the same amount of points as the bezier curve.

The red points marks the starting point and corresponds to the red point on the bezier curve.

The cameras position on the bezier curve is determined by the point on the linear path closest to the player/subject.

5.7. Camera Switching

The *CameraSwitchVolume* prefab is a box trigger used to place the camera onto a designated fixed point or bezier curve upon the subject entering the trigger volume.

The gameobject assigned to the *ThirdPersonTrackingCamera* script in the *follow* field is the object that activates the switching upon entering the trigger volume.

Setting Camera Intro to Zoom Intro plays a zooming animation (settings in the camera script) that helps retain the instant cut-style transition of camera switching with a bit of an animation to make it easier on the eye.

Alternatively No Intro option simply cuts the camera onto the new Curve or Point.

5.8. Using Other Camera Scripts

To use other camera scripts with this camera simply disable the *ThirdPersonTrackingCamera* script on the Camera gameobject.

This can be done through your own script or the *CameraSwitchTrigger* prefabs that can enable or disable the camera script on trigger enter.

6. Inspector properties

6.1. Bezier Editor

Select the CameraSpline to access these properties

6.1.1 Fixed Camera Spline

Free Rotate

Enables the Free rotate mode that rotates the camera toward the follow.

Read more at section 5.3.

Set FOV From Points

Uses FOV vales from points to control the FOV of the camera as it moves over the bezier curve.

Read more at section 5.4.

Roll Lock

Locks the camera Roll so that the camera is always upright.

Read mote at section 5.5.

[OPTIONAL] Reference path

Camera view follows the reference path according to the follows proximity to the reference path.

Read more at Section 5.6.

6.1.2 Bezier Curve Creator

Anchor Color

Color of the unselected Anchor points

Control Color

Color of the unselected control Points

Control Lines Color

Color of the lines that connect the control points to their anchor point

Bezier Color

Color of the main bezier curve

Anchor Size

Diameter of the anchor points in world units

Control Size

Diameter of the control points in world units

Create New

Deletes the Existing bezier curve and creates a new curve with two anchor points. (Undo available!)

Hide In Editor

Disables rendering of the curve in the Editor Window (leaves the name icon)

Hide In Game

Disables rendering the curve in Game View. Note: enabling this only works in Editor Game View as the rendering is an editor script. The curves wont render in game build.

Closed

Turns the Curve into a closed loop

Show Control Points

Renders Control points and their connections to the anchor points

Position Gizmo

Shows the position gizmo to be used to move the points (alternatively click + drag also moves the points)

Rotation Gizmo

Shows the rotation gizmo for rotating the anchor points

Automatic Control Points

Automatic control points are positioned automatically to make a smooth curve and automatically move according to the position of the anchor points.

Align Selected To Scene Camera

Places the selected point to the location of the scene/editor camera, used to easily position anchors to view the scene from the editors perspective.

Align Main Camera To Selected

Aligns the gameobject tagged “Main Camera” to the position and rotation of the anchor point in order to be able to see the point of view of the anchor point in the Game View.

6.1.2.1 Bezier Curve Creator – Point Selected

Selected Handle World Position

World position of the selected point

Selected Handle World Rotation

World Rotation of the selected point

Selected Handle Camera FOV

Only used if the *Set FOV from points* is enabled for this bezier.

The FOV value the camera will use when located at this point during gameplay.

6.2. Third Person Tracking Camera Script

The script on the *ThirdPersonTrackingCamera* Prefab.

Follow

The GameObject reference the Camera will follow. Necessary for the camera to function!

Follow Center

The Relative offset point the camera will track in relation to the follows origin point.

Adjust this if the origin point of the model isn’t the best focus point for the camera.

6.2.1 Automatic Rotate Mode Properties

These settings influence the camera behavior when its assigned to a Fixed point that allows rotation.

Rotation Smooth Speed

Lag the rotation to smooth out the movement. Influences free rotation mode for Fixed or Bezier follow.

Vertical Margin Percentage

Vertical margins that the subject of the camera will be in, that applies to the Fixed Point Free Rotate mode.

The Margins distance from the center of the screen is expressed in percentage ratio the edge of the screen (0 aka 0%) to the center of the screen (1.0 aka 100%).

The camera will start to move once the subject exits the margins in order to keep the subject within margins.

Visualize the margins by checking the **Draw Margins** option.

Horizontal Margin Percentage

Horizontal margins that the subject of the camera will be in, that applies to the Fixed Point Free Rotate mode.

The Margins distance from the center of the screen is expressed in percentage ratio the edge of the screen (0 aka 0%) to the center of the screen (1.0 aka 100%).

The camera will start to move once the subject exits the margins in order to keep the subject within margins.

Visualize the margins by checking the **Draw Margins** option.

6.2.2 Automatic Bezier Follow Mode

These settings influence the camera behavior when its following a bezier curve.

Movement Percentage Threshold

Margins of the screen that allow for subject movement without moving the camera in the Bezier Follow mode.

Expressed in percentage of the screen with 0 (0%) forcing the camera to find a position to keep the subject in the very center and 1.0 (100%) not moving the camera until the subject hits the borders of the screen.

This prevents an overly active camera since bezier curve doesn’t have full range of motion.

Visualize the margins by checking the **Draw Margins** option.

Bezier Iteration Count

The precision of the bezier curve is expressed in segmenting the curve recursively to find the closest matching point.

Higher values mean more precision but also exponentially more expensive computation times while lower values mean less precision but faster computation times.

Visualize the approximation with **Debug Draw Points Approximation**.

Bezier Follow Smooth Speed

Smooths the movement of the camera over the bezier curve, lower values mean smoother but slower movement and higher values mean faster but snappier movement.

Visualize the Smoothing with **Draw Bezier Approximation Points**.

6.2.3 Bezier Follow With Reference Path Mode

These settings influence the camera behavior when its following a bezier curve that has a Reference path assigned.

Smoooth Bezier Movement

Smooth the bezier movement in Reference path mode – the camera will smooth itself to the final position determined by the path.

Path Smooth Time

Time till the camera reaches its final point determined by the path.

6.2.4 Zoom Intro

These settings influence the camera behavior during camera switching if the Zoom intro is used.

Zoom Intro Time

The total duration of the zoom intro

Zoom Intro Starting FOV

The starting FOV of the zoom intro, this will linearly interpolate to the camera/bezier FOV.

6.2.5 Free Rotate Option

These Settings influence the camera when its following a bezier curve with the Free Rotate option checked.

Free Rotation Min Pitch

Minimum pitch the camera can achieve. Used to prevent the camera flipping under -90 deg. The value should be over -90.

Free Rotation Max Pitch

Minimum pitch the camera can achieve. Used to prevent the camera flipping over 90 deg. The value should be under 90.

6.2.6 Debug Draw

Helpful visualization

Draw Margin Lines

Used to visualize the margins for the Free Rotate mode. The visualization is persistent but the function is only used in Free Rotate.

Draw Bezier Curve Tracking Lines

Draw helpful lines to visualize the bezier movement

Blue – Camera Goal point

Red – Current Camera Point

Yellow – Camera Forward

Debug Draw Points Approximation

Visualize the approximation algorithm that determines the point on the curve. Useful when editing the iteration count, smooth speed and generally troubleshooting bezier behavior and ironing out problematic ares in your curve.

6.3. Switch Volume

Inspector variables in the Fixed Camera Trigger Script on the CameraSwitchTrigger prefab.

Spline

The reference to the Spline OR Fixed point the camera will switch to when the player/subject enters this switch volume.

The compatible references are *GameObjects* with either the *FixedCameraSpline* or *CameraFixedLocation* script (aka any script that inherits the CameraMovementConstraint script)

Both *CameraSpline* AND *CameraFixedLocation* prefabs are compatible with this field.

Camera Intro

Upon switching to the Spline/Point will the Zoom Intro

Trigger Cam Script

Set what the trigger will do to the *ThirdPersonTrackingCamera* Script.

Used to change camera scripts using the triggers.

None – no behavior

Enable – if the script is disabled it will enable it (and set the new constraint thats belongs to the trigger)

Disable – disables the script and clears constraints.

Draw Trigger

Draws the trigger in the scene during gameplay.

6.4. Camera Start

The Script that binds the camera to a Bezier curve or Fixed point on play start.

Start Bezier

The Bezier curve or Fixed Point that the camera starts with.

The compatible references are *GameObjects* with either the *FixedCameraSpline* or *CameraFixedLocation* script (aka any script that inherits the CameraMovementConstraint script)

Both *CameraSpline* AND *CameraFixedLocation* prefabs are compatible with this field.

Intro Type

Assign an intro animation when the game starts.